

# Backend Documentation (.NET API)

## Responsibility:

The backend is responsible for:

- Exposing a REST API
  - Enforcing business rules
  - Managing roles (Employee / Manager)
  - Persisting data
  - Ensuring data integrity for leave requests
- 

## Backend Architecture:

The backend follows a **layered architecture**:

Controllers → Services → Data (EF Core)

## Layers:

### Controllers:

- Expose REST endpoints
- Perform basic permission checks
- Delegate business logic to the service layer

### Examples:

- LeaveRequestsController
  - AuthController
- 

### Services:

- Contain **business logic**
- Isolate complex rules from controllers

Examples:

- Overlapping leave validation
  - Auto-rejection for requests longer than 15 days
- 

### **Data Layer:**

- EF Core + SQLite
  - Entities and DbContext
  - Initial seed data
- 

### **Authentication & Authorization**

#### **Approach:**

- **Simplified email-based login**
- Backend returns user information
- Frontend sends X-User-Id header with every request
  - X-User-Id: 1
  - X-User-Id: 2

#### **Design Decision**

JWT / ASP.NET Identity were intentionally avoided to keep the focus on business logic and architecture.

This approach is documented and suitable for the scope of this challenge.

---

#### **Business Rules (Backend)**

<b>Rule</b>	<b>Description</b>
Overlapping leave	Approved leave requests cannot overlap
> 15 days	Requests longer than 15 consecutive days are auto-rejected

<b>Rule</b>	<b>Description</b>
Roles	Only Managers can approve or reject requests
Deletion	Only the owner can delete a request if it is still Pending

All rules are enforced in the **service layer**, not in controllers.

---

## Main Endpoints

### Authentication:

<b>Method</b>	<b>Endpoint</b>	<b>Description</b>
POST	/api/auth/login	Login using email

---

### Leave Requests:

<b>Method</b>	<b>Endpoint</b>	<b>Permissions</b>
GET	/api/leaverequests	Employee: own / Manager: all
POST	/api/leaverequests	Employee
PUT	/api/leaverequests/{id}	Manager
DELETE	/api/leaverequests/{id}	Owner + Pending

---

## Backend Testing

- EF Core InMemory
- Tests covering critical rules:
  - Leave overlap prevention
  - Automatic rejection for long requests
  - Seed data (Employee + Manager)

## **CREATE LEAVE REQUEST EMPLOYEE:**

- **Header:** X-User-Id: 1
- **Method:** POST
- **URL:** <https://localhost:5001/api/LeaveRequests>
- **Body:**

```
{  
    "startDate": "2026-01-15",  
    "endDate": "2026-01-20",  
    "reason": "Medical"  
}
```

## **APPROVE OR REJECT REQUEST MANAGER:**

- **Header:** X-User-Id: 2
- **Method:** PUT
- **URL:** <https://localhost:5001/api/LeaveRequests/2>
- **Body:**

```
{  
    "status": 1  
}
```